Software Security Paper

CSC 424 Software Engineering II

Suzanne Moore

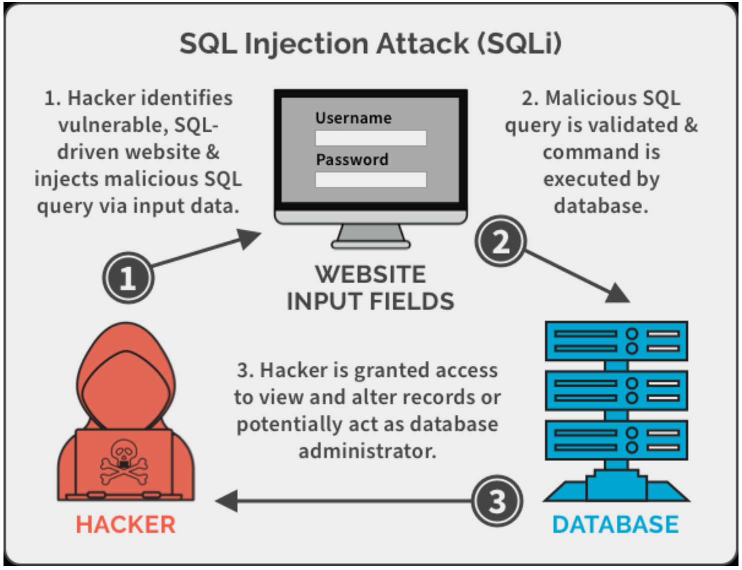
4-23-2021

Software security is important for all software development, not only does it provide protection for the software but for the user's input. Most software that is developed today, users' input includes personal data. Things like addresses, names, date of birth, security questions, and even credit card, are all pieces of data that people would certainly want private and not released to anyone. According to The Importance of Security to a Software Development Company [6]

> This is why security is so important: ensuring that data is secure obviously builds and strengthens the relationship between the business and its consumer. Consumers will definitely keep their business with a software development company if they know that their data is secure [6].

Any type of data that is being collected or stored, it should be protected from fraud and unauthorized access by internal or external sources. This not only protects a company or code quality but can cause damage to future consumer confidence, finical lost, and reputation. Providing code security will limit security risk by removing vulnerability in code that is created for a specific task. To provided software security developers should focus on how they can write the most secure code possible due to the languages and tools used for a project. Secure coding practices should be incorporated in all stages of the development process. A few ways to secure code are validating input for untrusted data, testing, maintenance, and keeping things simple.

Software security should be addressed when describing the development of the project all the way to end. One common software security attacks are a SQL injection. SQL injections are vulnerabilities that an attacker will find in the user input in a webpage. Since SQL is a programming language used to manage data stored in a database, you can use it to delete data and modify it. If an SQL injection attack were to happen may concerns can happen to data and information stored. With an SQL injection an attacker can find credentials of other users, delete records from a database, gain access to the operating system using the database server. These attacks will cause serious consequences so an SQL injection test should be required to prevent such attacks. According to Bird [4]


SQL Injection Attack (SQLi)
1. Hacker identifies vulnerable, SQL-driven website & injects malicious SQL query via input data.
Username
Password
2. Malicious SQL query is validated & command is executed by database.
1
2
WEBSITE INPUT FIELDS
3. Hacker is granted access to view and alter records or potentially act as database administrator.
3
HACKER
DATABASE

[1]

> Luckily, SQL injection is also easy to prevent. You simply need to parameterize your SQL statements, making it clear to the SQL interpreter which parts of a SQL statement make up the command and which parts are data. OWASP has a Cheat Sheet that explains

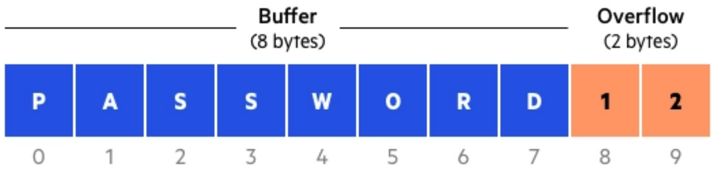
> how to parameterize queries in Java (using prepared statements or Hibernate) and in other languages [4].

To protect a webpage from an SQL injection you can use SQL parameters which are values that are added to an SQL query at run time. Other ways to prevent an SQL injection attack is to validate the user input, sanitize the data by limiting individual characters, and monitor SQL statements. Since SQL injection are not too uncommon there is now a few SQL injection tools that can be used such as SQLMap, BBQSQL, jSQL Injection, and Whitewidow.

Another software security vulnerability is buffer overflow. A buffer overflow is when a program attempts to put more data in a buffer than it can hold. According to Wikimedia Foundation [8]

> A buffer overflow occurs when data written to a buffer also corrupts data values in memory addresses adjacent to the destination buffer due to insufficient bounds checking. This can occur when copying data from one buffer to another without first checking that the data fits within the destination buffer [8].

When the buffer overflows it can overwrite data, allow the program to not act as it should, and could cause memory access errors then crash the program. Since C++ is a very a language that allows the programmer to do many things with memory, it is mostly associated with buffer overflows. The C++ language is so flexible that there is no built-in safeguard against accessing or overwriting data in memory. If a log in is requesting a password and is using 8 bytes, if 10 bytes is used the 2 bytes would cause the buffer to be overflown.



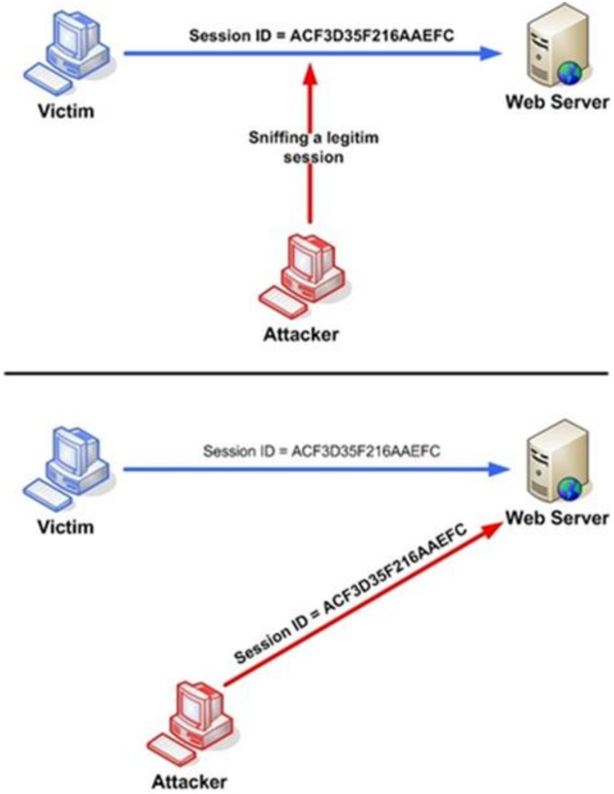
[5]

Buffer overloads can lead to not only system crashes but could cause the program to put into an infinite loop. There are ways to provided buffer overflow protection by choosing a programming language other than assembly and C++, which are usually programming languages that take part in buffer overflow. When it come down to the design of a program, adding in safe libraries and avoid libraries that are not bounds checked. Buffer overflow protection systems can be used to find most common buffer overflow by checking if a stack has not been altered during a run of a program [8]. The current systems that can be used for buffer flow protection are Libsafe and StackGuard. Testing the software one of the best methods of checking for buffer overflow while patching up bugs at the same time. According to Wikimedia Foundation "Once a potential buffer overflow is detected, it must be patched; this makes the testing approach useful for software that is in development, but less useful for legacy software that is no longer maintained or supported" [8].

Another type of software security issue with today's technology is session hijacking. According to Wikimedia Foundation "In computer science, session hijacking, sometimes also known as cookie hijacking is the exploitation of a valid computer session—sometimes also called a session key—to gain unauthorized access to information or services in a computer system" [9]. Session hijacks are usually within busy networks with high numbers of active communication sittings. High network use provides the attacker with many sessions to manipulate and protection since many active sessions are already on the server. According to Arampatzis [2]

> When cybercriminals have hijacked a session, they can do virtually anything that the legitimate user was authorized to do during the active session. The most severe examples include transferring money from the user's bank account, buying merchandise from web stores, accessing personally identifiable information (PII) for identity theft, and even stealing data from company systems [2].

Session hijack can manipulate the vulnerability of a system by accessing user's session ID's by using webpages that allow session ID's form URLs. When the session hijacking attacker get access to the session of user, they take the valid session ID, which is used to get into to system and get the data they need. There are different types of session hijacking, a few common session hijackings are session sniffing, client-side attacks, and blind hijacks. While there are many ways for an attacker to create a session hijacking attack, there are way to prevent this attack from accruing. A few ways to prevent session hijacking are provided long number as a session key, prevent session fixation, use long and random session IDs, time-out inactive sessions, and perform secondary checks. Another way to prevent these attacks are use HTTPS. Using HTTPS will help prevent the attacker from stopping the plaintext session ID, even if they are tracking previous activity by a user.  A programmer can also use HSTS (HTTP Strict Transport Security) to help improve security [3].


Session ID = ACF3D35F216AAEFC
Victim
Web Server
Sniffing a legitim session
Attacker
Session ID = ACF3D35F216AAEFC
Victim
Web Server
Session ID = ACF3D35F216AAEFC
Attacker

To sum, there are many ways to increase software security based on what the program is design and what input is required. No matter what you think, all software is prone to have security threats, breaches of data, or buffer overflow. Before starting any software development, it is important to be aware of the security requirement the team has planned out to prevent an attack or data loss. All team members are responsible for the software security and should provide a test plan to check all security requirements. This can be allowing on team member to be the lead on project security to make sure security issues and being developed during the built of the project. There has been an increase on software security testing tools to help analyze how the code is written and interacts with the environment. This is useful to identify a programs weakness or a flaw in the design of the project. As a program is developed from the beginning to finished it should be updated on the design, implementation, and security. Security requirements should be listed as one of the first topics to make sure the project provides high quality code that provided high level of security. The high code quality of security is essential as having a software project that is functional, there are numerous ways to test code for vulnerabilities to have a software with limited security risk.

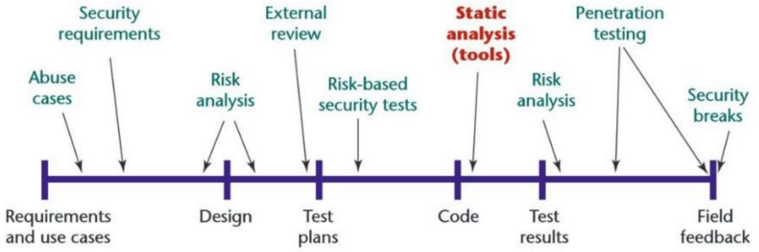


Figure 1. The software development life cycle. Throughout this series, we'll focus on specific parts of the cycle; here, we're examining static analysis.

[7]

# References

[1] Abeythissa, I. (2019, November 20). *Blind SQL Injection | Triggering Conditional Response | Part 1 | Ishara Abeythissa*. https://isharaabeythissa.medium.com/blind-sql-injection-triggering-conditional-response-8b49c6f75512.

[2] Arampatzis, A. (2021, April 12). What is Session Hijacking? Venafi. https://www.venafi.com/blog/what-session-hijacking.

[3] Banach, Z. (2019, August 22). What Is Session Hijacking: Your Quick Guide to Session Hijacking Attacks. Netsparker. https://www.netsparker.com/blog/web-security/session-hijacking/.

[4] Bird, J. (2015, December 14). *10 Steps to Secure Software*. dzone.com. https://dzone.com/articles/10-steps-to-secure-software.

[5] Buffer Overflow Attack. (n.d.). https://www.imperva.com/learn/application-security/buffer-overflow/.

[6] The Importance of Security to a Software Development Company. Vestra Inet. (2018, August 17). https://vestrainet.com/the-importance-of-security-to-a-software-development-company.html.

[7] McGraw , G. (2004, March 31). *Software security*. Software Integrity Blog. https://www.synopsys.com/blogs/software-security/software-security/.

[8] Wikimedia Foundation. (2021). Buffer overflow. Wikipedia. https://en.wikipedia.org/wiki/Buffer_overflow#:~:text=A%20buffer%20overflow%20occurs%20when,fits%20within%20the%20destination%20buffer.

[9] Wikimedia Foundation. (n.d.). Session hijacking. Wikipedia. https://en.wikipedia.org/wiki/Session_hijacking.